REMARKS/ARGUMENTS

Claims 9-12 have been amended to further clarify the subject matter regarded as the invention. New Claims 22-23 have been added. Thus, claims 1-23 are now pending.

In the Office Action, the Examiner has rejected claims 1-21 under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,633,923 (*Kukura et al.)* in view of "x-kernel tutorial by Peterson, Davie and Bavier", published on January 1996, *Peterson.* This rejection is fully traversed below.

The present application relates to techniques for transmission of messages between Object Request Brokers that typically operate in distributed object-oriented computing environments. The techniques provide a mechanism for generating and transmitting message fragments (or sub-messages) form an original message in accordance with one aspect of the invention. In one embodiment, a fragment-offset is provided. The fragment-offset, among other things, can be used to determine the location of data bytes in the message fragments (or sub-messages) with respect to the original message. It should also be noted that the fragment-offset can be updated to indicate the current offset with respect to the original message when a sub-message is constructed (Specification, page 10, lines 3-9).

Unlike conventional techniques, a significant amount of computations or bookkeeping (e.g., number of fragments constructed, total number of bytes constructed, etc.) is not required to construct message fragments. As will also appreciated, sub-messages can have header that vary in size. Similarly, the data portions of the sub-messages can vary in size. As a result, the techniques provides an elegant mechanism to determine the location of bytes in sub-messages for various messages that are constructed and transmitted between Object Request Brokers in a distributed object-oriented computing environments (Specification, page 10, lines 9-17).

Claim 1 pertains to a method of sending a message from a first common Object Request Broker to a second common Object Request Broker operating in a distributed object oriented environment. As such, claim 1 recites: determining whether a message is to be fragmented in two or more sub-messages, initiating construction of a sub-

message, initializing an offset-variable to zero, determining whether there is a need to know the position of a byte of the sub-message with respect to the message, reading the offset-variable, completing construction of the sub-message based on the offset-variable, updating the offset-variable, and sending a constructed sub-message from the first common Object Request Broker to a second common Object Request Broker.

In the Office Action, the Examiner has noted that *Kukura et al.* is silent about: determining whether a message is to be fragmented in two or more sub-messages, initializing an offset, reading the offset variable, completing construction of the sub-message based on the offset-variable, and updating the offset-variable (Office Action, page 4). Initially, it is respectfully submitted that *Peterson* cannot possibly overcome the serious deficiencies of *Kukura et al.* Accordingly, it is respectfully submitted that the Examiner has not made a prima facie case of obviousness.

Moreover, it is respectfully submitted that *Kukura et al.* does not pertain to a method for sending a message from a first common Object Request Broker to a second common Object Request Broker operating in a distributed object oriented environment. Instead, *Kukura et al.* pertains to dynamic configuration of interceptors (*Kukura et al.,* title). The interceptors are used for a binding process. A binding is the set of interceptors involved in communication between a particular client and a particular object or set of objects. Different interceptor and binding interfaces are involved in the client and server roles that make up an invocation (*Kukura et al.,* col. 13, 16-20). In CORBA, the binding process is initiated by a client application attempting to make an invocation using an object reference to which no appropriate binding already exists. The target object reference identifies the server application with which a binding will be established. The binding that is established represents the negotiated protocol and services that allow the client to make invocations on the target object, and any shared state required by the protocol and services at the client and server (*Kukura et al.,* col. 5, 38-46).

Accordingly, it is respectfully that *Kukura et al.* does NOT teach or suggest determining whether a message is to be fragmented in two or more sub-messages, initiating construction of a sub-message, initializing an offset-variable to zero, determining whether there is a need to know the position of a byte of the sub-message with respect to the message, reading the offset-variable, completing construction of the sub-message based on the offset-variable, updating the offset-variable, and sending a

constructed sub-message from the first common Object Request Broker to a second common Object Request Broker in the context of the invention.

Furthermore, it is respectfully that *Peterson* does NOT teach or suggest these features. *Peterson* pertains to a tutorial on writing x-kernel protocols. The x-kernel provides an object-based framework for implementing protocols such as TCP or IP networking protocols (*Peterson*, page 3). *Peterson,* however, does not teach or suggest a method for sending a message from a first common Object Request Broker to a second common Object Request Broker in a distributed object oriented environment. As such, *Peterson* **cannot** possibly teach these features in the context of the claimed invention. Nevertheless, some of the claimed features that are not taught by *Peterson* will be discussed below.

It should be noted that *Peterson* describes BLAST as an X-kernel protocol that fragments large messages into MTU-sized packets at the sender, and reassembles the fragments back into the complete message at the receiver (*Peterson*, page 31-32). However, contrary to the Examiner's assertion, it is respectfully submitted that the fragmentation (9.1) and reassembly (9.2) do NOT teach or suggest initializing an offset-variable to zero when it is determined that the message is to be fragmented into two or more sub-messages, determining whether there is a need to know the position of a byte of the sub-message with respect to the message, and completing construction of the sub-message based on the offset-variable. In addition, it should be noted that in adding and striping of headers (3.1) *Peterson* teaches adding and striping headers of a message as it goes up and down a protocol graph (*Peterson,* page 10). Thus, contrary to the Examiner's assertion, adding and striping of headers (3.1) does not teach reading the offset-variable when there is a need to know the position of a byte of the sub-message with respect to the message.

Accordingly, it is respectfully submitted that claim 1 is patentable over *Kukura et al.* and *Peterson,* taken alone, or in any proper combination. In addition, claims that are dependent on claim 1 are patentable for at least this reason. Moreover these dependent claims recite additional features that render them patentable for additional reasons. For example, claims 2, 3 and 4 further recite additional updating features. Contrary to the Examiner's assertions, it is respectfully submitted that Col. 33, 37, and 38 of *Kukura et al.* do not teach or suggest these additional features.

Claim 12 also pertains to a method for sending a message from a first common Object Request Broker to a second common Object Request Broker. Moreover, claim 12 recites similar features as those recited in claim 1. Accordingly, it is respectfully submitted that claim 12 and its dependent claims are also patentable for similar reasons. Although claim 17 is directed to a computer readable media, it also recites similar features as discussed above with respect to claim 1. Thus, it is respectfully submitted that claim 7 and its dependent claims are also patentable for similar reasons.

Claim 9 pertains to a computing system that operates in a distributed object-oriented environment. Claim 9, among other things, recites a message fragment offset-variable which is provided for a first common Object Request Broker. The message fragment offset-variable indicates a position of a byte of a sub-message with respect to the entire message. This position can be determined based on the message fragment offset-variable by subtracting the length of the header of the sub-message from the length of another sub-message immediately preceding the sub-message, and then adding the result of the subtraction to the value of the message fragment offset-variable.

As noted above, *Kukura et al.* and *Peterson* taken alone, or in any proper combination, do not teach or suggest a message fragment offset-variable which is provided for a first common Object Request Broker in the context of the invention. In addition, neither *Kukura et al,* nor *Peterson* teach of suggest determining the position on the message fragment offset-variable by subtracting the length of the header of the sub-message from the length of another sub-message immediately preceding the sub-message, and then adding the result of the subtraction to the value of the message fragment offset-variable. Thus, it is respectfully submitted that claim 9 and its dependent claims are also patentable for at least these reasons.

Based on the foregoing, it is submitted that claims 1-21 are patentably distinct over the cited art of record. Additional limitations recited in the independent claims or the dependent claims are not further discussed because the limitations discussed above are sufficient to distinguish the claimed invention from the cited art. Accordingly, Applicant believes that all pending claims are allowable and respectfully requests a Notice of Allowance for this application from the Examiner.

Applicants hereby petition for an extension of time which may be required to maintain the pendency of this case, and any required fee for such extension or any

further fee required in connection with the filing of this Amendment is to be charged to Deposit Account No. 500388 (Order No. SUN1P807). Should the Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

Respectfully submitted,
BEYER WEAVER & THOMAS, LLP

R. Mahboubian
Reg. No. 44,890

P.O. Box 778
Berkeley, CA 94704-0778